ENCRYPTION APPARATUS AND METHOD IN A WIRELESS COMMUNICATIONS SYSTEM

PRIORITY

This application claims priority under 35 U.S.C. § 119 to an application entitled "Encryption Apparatus and Method in a Wireless Communications System" filed in the Korean Industrial Property Office on October 8, 2002 and assigned Serial No. 2002-61179, the contents of which are incorporated herein by reference.

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates generally to a wireless communications system, and in particular, to an encryption apparatus and method for implementing confidentiality and integrity algorithms in a wireless communications system.

2. Description of the Related Art

As the first generation analog encryption system has evolved into the second generation digital encryption system, more advanced encryption techniques have been used. The current third generation encryption system provides encryption service for multimedia service i.e., audio and video information. Thus, the importance of encryption has increased in order to provide confidentiality to voice signals, multimedia service, and user data. An integrity algorithm is required to authenticate control signals between mobile terminals in a wireless communication system and a network. The 3rd Generation Project Partnership (3GPP) has selected the KASUMI algorithm as the f8 confidentiality and f9 integrity algorithms for a third generation system based on a Global

System for Mobile communication (GSM) core network, and a Universal Mobile Telecommunication System (UMTS).

FIG. 1 is a block diagram illustrating an example of a conventional KASUMI algorithm. Referring to FIG. 1, KASUMI is an 8-round Feistel unit cipher that provides a 64-bit output ciphertext from a 64-bit input plaintext with 8-round encryption. The 64-bit input signal is divided into a 32-bit signal L_0 and a 32-bit signal R_0. FLi units ($1 \leq i \leq 8$) 110 to 180 and FOi units ($1 \leq i \leq 8$) 210 to 280 encrypt the signals L_0 and R_0 under corresponding encryption keys KL_i ($1 \leq i \leq 8$), KO_i ($1 \leq i \leq 8$), and KI_i ($1 \leq i \leq 8$) and output the 64-bit ciphertext.

Encryption in accordance with FIG. 1 occurs in the following manner. An FL1 unit 110 encrypts the input 32-bit signal L_0 with an encryption key KL_1 and outputs a ciphertext L_{01}. An FO1 unit 210 encrypts the 32-bit ciphertext L_{01} with encryption keys KO_1 and KI_1 and outputs a ciphertext L_{02}. An Exclusive-OR operation is performed to logically "exclusive OR" the ciphertext L_{02} and the 32-bit signal R_0 to provide a 64-bit ciphertext. This encryption occurs eight times and a final 64-bit ciphertext is generated in the KASUMI.

FIG. 2A is a block diagram illustrating an example of FOi units. Referring to FIG. 2A, FOi denotes an ith FO unit. The FOi unit comprises a plurality of $Fl_{i,j}$ sub-ciphers ($1 \leq i \leq 3$, $1 \leq i \leq 3$) to provide 3-rounds of encryption. Here, the operation of the FO1 unit 210 will be described by way of example. The 32-bit input signal is divided into two 16-bit signals L_0 and R_0. An Exclusive-OR operation is performed to logically "exclusive OR" the 16-bit signal L_0 and a 16-bit sub-encryption key $KO_{1,1}$, to provide a signal L_1. A $Fl_{1,1}$ sub-cipher 201 encrypts the signal L_1 with a 16-bit sub-encryption key $KI_{1,1}$ and outputs a signal L_{1D}. Meanwhile, a first delay (D_1) 10 delays the 16-bit signal R_0, which is equivalent to the signal R_1, in order to synchronize the 16-bit signal R_0 with the signal L_{1D} and output a delayed signal R_{1D}. For a second-round of

encryption, an Exclusive-OR operation is performed to logically "exclusive OR" the 16-bit signal R_{1D} and a 16-bit sub-encryption key $KO_{1,2}$ to provide a signal L_2. A $Fl_{1,2}$ sub-cipher 203 encrypts the signal L_2 with a 16-bit sub-encryption key $KI_{1,2}$ and outputs a signal L_{2D}. Meanwhile, an Exclusive-OR operation is performed to logically "exclusive OR" the 16-bit signal R_{1D} and the signal L_{1D}, to provide a signal R_2. A second delay (D_2) 20 delays the signal R_2 in order to synchronize the signal R2 with the signal L_{2D} and output a delayed signal R_{2D}. For a third-round of encryption, an Exclusive-OR operation is performed to logically "exclusive OR" the 16-bit signal R_{2D} and a 16-bit sub-encryption key $KO_{1,3}$, resulting in a signal L_3. A $Fl_{1,3}$ sub-cipher 205 encrypts the signal L_3 with a 16-bit sub-encryption key $KI_{1,3}$ and outputs a signal L_{3D}. Meanwhile, an Exclusive-OR operation is performed to logically "exclusive OR" the 16-bit signal R_{2D} and the signal L_{2D} to provide a signal R_3. A third delay (D_3) 30 delays the signal R_3 in order to synchronize the signal R_3 with the signal L_{3D} and output a delayed signal R_{3D}. An Exclusive-OR operation is performed to logically "exclusive OR" the 16-bit signal R_{3D} and the signal L_{3D}, to provide a signal R_4. The 16-bit signal R_4 is operated with the 16-bit signal R_{3D} ($=L_4$), resulting in a 32-bit ciphertext $L_4 // R_4$.

The FO1 unit uses the three delays 10, 20 and 30 to synchronize to the output timings of the sub-ciphers 201, 203 and 205.

FIG. 2B is a block diagram illustrating another example of the FOi units. Referring to FIG. 2B, a FOi unit comprises a plurality of $Fl_{i',j'}$ sub-ciphers ($1 \leq i' \leq 3$ $1 \leq j' \leq 3$), for 3-rounds of encryption. Here, the FO1 unit 210 will be described by way of example. The 32-bit input signal is divided into two 16-bit signals $L_{0'}$ and $R_{0'}$. An Exclusive-OR operation is performed to logically "exclusive OR" the 16-bit signal $L_{0'}$ and a 16-bit sub-encryption key $KO_{1,1}$, to provide a signal $L_{1'}$. A $Fl_{1',1'}$ sub-cipher 211 encrypts the signal $L_{1'}$ with the 16-bit sub-encryption key $KI_{1,1}$ and outputs a signal $L_{1D'}$. Meanwhile, a fourth delay

(D_4) 40 delays the 16-bit signal R_0, (=R_1,) and outputs a delayed signal R_{1D},. An Exclusive-OR operation is performed to logically "exclusive OR" the signals L_{1D}, and R_{1D}, to provide a signal L_2,. Simultaneously, an Exclusive-OR operation is performed to logically "exclusive OR" the 16-bit signal R_0, and a 16-bit sub-encryption key $KO_{1,2}$, to provide a signal R_2,. A $Fl_{1},_{2}$, sub-cipher 213 encrypts the signal R_2, with a 16-bit sub-encryption key $KI_{1,2}$ and outputs a signal R_{2D},. An Exclusive-OR operation is performed to logically "exclusive OR" the signals L_2, and R_{2D}, to provide a signal R_3,. Another Exclusive-OR operation is performed to logically "exclusive OR" the signal L_2, and a 16-bit sub-encryption key $KO_{1,3}$, to provide a signal L_3,. A $Fl_{1},_{3}$, sub-cipher 215 encrypts the signal L_3, with a 16-bit sub-encryption key $KI_{1,3}$ and outputs a signal L_{3D},. Meanwhile, a fifth delay (D_5) 50 delays the signal R_3, and outputs a delayed signal R_{3D},. An Exclusive-OR operation is performed to logically "exclusive OR" the signals L_{3D}, and R_{3D}, to provide a 16-bit signal L_4,. The 16-bit signal L_4, is operated with the 16-bit signal R_{3D}, (=R_4,), resulting in a 32-bit ciphertext L_4,//R_4,.

The above advanced FOi unit uses the two delays 40 and 50 to synchronize to the output timings of the Fl sub-ciphers 211 and 215. However, due to the use of the delays, a large chip capacity is required.

FIG. 3 is a block diagram illustrating an example of the $Fl_{i,j}$ sub-ciphers illustrated in FIGs. 2A and 2B. By way of example, the $Fl_{1,1}$ sub-cipher 201 will be described below. Referring to FIG. 3, the 16-bit input signal is divided into a 9-bit signal RL_0 and a 7-bit signal RR_0. An SBox91 (S91) operator 310 generates a 9-bit signal y0, y1, . . ., y8 from the input signal RL_0 using

$$
\begin{aligned}
y0 &= x0x2\oplus x2\oplus x2x5\oplus x5x6\oplus x0x7\oplus x1x7\oplus x2x7\oplus x4x8\oplus x5x8\oplus x7x8\oplus 1\\
y1 &= x1\oplus x0x1\oplus x2x3\oplus x0x4\oplus x1x4\oplus x0x5\oplus x3x5\oplus x6\oplus x1x7\oplus x2x7\oplus x5x8\oplus 1\\
y2 &= x1\oplus x0x3\oplus x3x4\oplus x0x5\oplus x2x6\oplus x3x6\oplus x5x6\oplus x4x7\oplus x5x7\oplus x6x7\oplus x8\oplus x0x8\oplus 1\\
y3 &= x0\oplus x1x2\oplus x0x3\oplus x2x4\oplus x5\oplus x0x6\oplus x1x6\oplus x4x7\oplus x0x8\oplus x1x8\oplus x7x8\\
y4 &= x0x1\oplus x1x3\oplus x4\oplus x0x5\oplus x3x6\oplus x0x7\oplus x6x7\oplus x1x8\oplus x2x8\oplus x3x8\\
y5 &= x2\oplus x1x4\oplus x4x5\oplus x0x6\oplus x1x6\oplus x3x7\oplus x4x7\oplus x6x7\oplus x5x8\oplus x6x8\oplus x7x8\oplus 1\\
y6 &= x0\oplus x2x3\oplus x1x5\oplus x2x5\oplus x4x5\oplus x3x6\oplus x4x6\oplus x5x6\oplus x7\oplus x1x8\oplus x3x8\oplus x5x8\oplus x7x8\\
y7 &= x0x1\oplus x0x2\oplus x1x2\oplus x3\oplus x0x3\oplus x2x3\oplus x4x5\oplus x2x6\oplus x3x6\oplus x2x7\oplus x5x7\oplus x8\oplus 1\\
y8 &= x0x1\oplus x2\oplus x1x2\oplus x3x4\oplus x1x5\oplus x2x5\oplus x1x6\oplus x4x6\oplus x7\oplus x2x8\oplus x3x8
\end{aligned}
$$

. . . . . (1)

A ZE1 unit 320 receives the signal RR_0, adds two zeroes to the Most Significant Bit (MSB) of the signal RR_0, and outputs a 9-bit signal. An Exclusive-OR operation is performed to logically "exclusive OR" the outputs of the S91 operator 310 and the ZE1 unit 320 to provide a 9-bit signal RL_1. Another Exclusive-OR operation is performed to logically "exclusive OR" the signal RL_1 and a 9-bit sub-encryption key $KI_{1,1,2}$, to provide a 9-bit signal RL_2.

A TR1 unit 330 removes two zero bits from the MSBs of the 9-bit signal RL_1. An SBox71 (S71) operator 340 generates a 7-bit signal y0, y1, . . ., y6 from the input signal RR_0 (=RR_1) by

$$
\begin{aligned}
y0 &= x1x3\oplus x4\oplus x0x1x4\oplus x5\oplus x2x5\oplus x3x4x5\oplus x6\oplus x0x6\oplus x1x6\oplus x3x6\oplus x2x4x6\oplus x1x5x6\\
&\qquad \oplus x4x5x6\\
y1 &= x0x1\oplus x0x4\oplus x2x4\oplus x5\oplus x1x2x5\oplus x0x3x5\oplus x6\oplus x0x2x6\oplus x3x6\oplus x4x5x6\oplus 1\\
y2 &= x0\oplus x0x3\oplus x2x3\oplus x1x2x4\oplus x0x3x4\oplus x1x5\oplus x0x2x5\oplus x0x6\oplus x0x1x6\oplus x2x6\oplus x4x6\oplus 1\\
y3 &= x1\oplus x0x1x2\oplus x1x4\oplus x3x4\oplus x0x5\oplus x0x1x5\oplus x2x3x5\oplus x1x4x5\oplus x2x6\oplus x1x3x6\\
y4 &= x0x2\oplus x3\oplus x1x3\oplus x1x4\oplus x0x1x4\oplus x2x3x4\oplus x0x5\oplus x1x3x5\oplus x0x4x5\oplus x1x6\oplus x3x6\\
&\qquad \oplus x0x3x6\oplus x5x6\oplus 1\\
y5 &= x2\oplus x0x2\oplus x0x3\oplus x1x2x3\oplus x0x2x4\oplus x0x5\oplus x2x5\oplus x4x5\oplus x1x6\oplus x1x2x6\oplus x0x3x6\\
&\qquad \oplus x3x4x6\oplus x2x5x6\oplus 1\\
y6 &= x1x2\oplus x0x1x3\oplus x0x4\oplus x1x5\oplus x3x5\oplus x6\oplus x0x1x6\oplus x2x3x6\oplus x1x4x6\oplus x0x5x6
\end{aligned}
$$

. . . . . (2)

An Exclusive-OR operation is performed to logically "exclusive OR" the

outputs of the TR1 330 and the S71 operator 340 via a sub-encryption key $KI_{1,1,1}$, to provide a 7-bit signal RR_2.

A SBox92 (S92) operator 350 generates a 9-bit signal y0, y1, . . ., y8 from the signal RL_2 by Eq. (1). A ZE2 unit 360 receives the signal RR_1, adds two zeroes to the MSB of the signal RR_1, and outputs a 9-bit signal. An Exclusive-OR operation is performed to logically "exclusive OR" the outputs of the S92 operator 350 and the ZE2 unit 360 to provide a 9-bit signal RL_3. A TR2 unit 370 removes two zero bits from the MSBs of the 9-bit signal RL_3. A SBox72 (S72) operator 380 generates a 7-bit signal y0, y1, . . ., y6 from the input signal RR_2 (=RR_3) using Eq. (2). Another Exclusive-OR operation is performed to logically "exclusive OR" the outputs of the TR2 370 and the S72 operator 380 to provide a 7-bit signal RR_4.

The 9-bit signal RL_3 (=RL_4) and the 7-bit signal RR_4 are operated, resulting in a 16-bit ciphertext $RL_4//RR_4$.

As described above, the S91 operator 310 and the S92 operator 350 each sequentially perform an AND operation to perform a logical "AND" and an exclusive-OR operation to perform a logical "Exclusive-OR" using Eq. (1), to thereby generate an output signal y0, y1, . . ., y8. Similarly, the S71 operator 340 and the S72 operator 380 sequentially perform an AND operation to perform a logical "AND" and an exclusive-OR operation to perform a logical "Exclusive-OR" using Eq. (2), to thereby generate an output signal y0, y1, . . ., y6. Consequently, the encryption speed is decreased. Moreover, a gate delay involved in the operations of the S91, S92, S71 and S72 operators 310, 350, 340, and 360 gradually increases glitch.

SUMMARY OF THE INVENTION

It is, therefore, an object of the present invention to provide an

encryption method for generating a ciphertext bit stream of length 2n from a plaintext bit stream of length 2n.

It is another object of the present invention to provide an encryption apparatus for generating a ciphertext bit stream of length 2n from a plaintext bit stream of length 2n.

To achieve the above objects, in an encryption method for dividing a first plaintext bit stream of length 2n into first and second sub-bit streams of length n, dividing a second plaintext bit stream of length 2n into third and fourth sub-bit streams of length n, and generating a ciphertext bit stream of length 2n from the first, second, third and fourth sub-bit streams using 2-rounds of encryption, first and second ciphertext bit streams of length n are generated by encrypting the first and second sub-bit streams with predetermined first encryption codes $KO_{1,1}$, $KO_{1,2}$, $KO_{1,3}$, $KI_{1,1}$, $KI_{1,2}$, and $KI_{1,3}$, the second ciphertext bit stream being output with a predetermined time delay from the first ciphertext bit stream, in a first-round encryption. A first operated ciphertext bit stream is generated by performing a logical exclusive-OR-operation on the first ciphertext bit stream and the third sub-bit stream, and a second operated ciphertext bit stream is operated by performing a logical exclusive-OR-operation on the second ciphertext bit stream and the fourth sub-bit stream. In a second-round of encryption, third and fourth ciphertext bit streams of length n are generated by encrypting the first operated ciphertext bit stream and the second operated ciphertext bit stream with predetermined second encryption codes $KO_{2,1}$, $KO_{2,2}$, $KO_{2,3}$, $KI_{2,1}$, $KI_{2,2}$, and $KI_{2,3}$ and the third and fourth ciphertext bit streams are concurrently output.

In an encryption apparatus for dividing a first plaintext bit stream of length 2n into first and second sub-bit streams of length n, dividing a second plaintext bit stream of length 2n into third and fourth sub-bit streams of length n,

and generating a ciphertext bit stream of length 2n from the first, second, third and fourth sub-bit streams using 2-rounds of encryption, a first ciphering unit receives the first and second sub-bit streams, and generates first and second ciphertext bit streams of length n by encrypting the first and second sub-bit streams with predetermined first encryption codes $KO_{1,1}$, $KO_{1,2}$, $KO_{1,3}$, $KI_{1,1}$, $KI_{1,2}$, and $KI_{1,3}$. Here, the second ciphertext bit stream is output with a predetermined time delay from the first ciphertext bit stream. An operating unit generates a first operated ciphertext bit stream by performing a logical exclusive-OR operation on the first ciphertext bit stream and the third sub-bit stream, and generates a second operated ciphertext bit stream by performing a logical exclusive-OR-operation on the second ciphertext bit stream with the fourth sub-bit stream. A second ciphering unit receives the first operated ciphertext bit stream and the second operated ciphertext bit stream having the predetermined time delay, generates third and fourth ciphertext bit streams of length n by encrypting the first operated ciphertext bit stream and the second operated ciphertext bit stream with predetermined second encryption codes $KO_{2,1}$, $KO_{2,2}$, $KO_{2,3}$, $KI_{2,1}$, $KI_{2,2}$, and $KI_{2,3}$ and concurrently outputs the third and fourth ciphertext bit streams.

BRIEF DESCRIPTION OF THE DRAWINGS

The above and other objects, features and advantages of the present invention will become more apparent from the following detailed description when taken in conjunction with the accompanying drawings in which:

FIG. 1 is a block diagram illustrating an example of a conventional KASUMI algorithm;

FIG. 2A is a block diagram illustrating an example of FOi units illustrated in FIG. 1;

FIG. 2B is a block diagram illustrating another example of the FOi units illustrated in FIG. 1;

FIG. 3 is a block diagram illustrating an example of $Fl_{i,j}$ sub-ciphers illustrated in FIGs. 2A and 2B;

FIG. 4 is a block diagram illustrating an example of a KASUMI algorithm according to the present invention;

FIG. 5 is a block diagram illustrating an example of SLIMFOi units illustrated in FIG. 4 according to the present invention; and

FIG. 6 is a block diagram illustrating an example of $Fl_{i,j}$ sub-ciphers illustrated in FIG. 5 according to the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

An embodiment of the present invention will be described herein below with reference to the accompanying drawings. Also, a description of known functions and configurations have been omitted for conciseness.

A KASUMI algorithm according to the present invention is a ciphering algorithm used as the f8 confidentiality and f9 integrity algorithms. The f8 confidentiality algorithm encrypts a plaintext signal having a predetermined number of bits by exclusive-OR-operating the plaintext with an encryption key and decrypts a ciphertext by exclusive-OR-operating the ciphertext with the encryption key. The f9 integrity algorithm derives a message authentication code from a received signal. The KASUMI algorithm, as previously discussed, has emerged as a significant issue to confidentiality and integrity.

FIG. 4 is a block diagram illustrating an example of a KASUMI algorithm according to the present invention. Referring to FIG. 4, a KASUMI of the present invention provides a 64-bit output ciphertext from a 64-bit input plaintext using first, second and third encryption keys. The 64-bit input signal is divided into a 32-bit signal L_0 and another 32-bit signal R_0. FLi units ($1 \leq i \leq 8$) 410 to 480 and SLIMFOi units ($1 \leq i \leq 4$) 510 to 540 are used to encrypt the signals L_0

and R_0 using corresponding encryption keys KO_i ($1 \leq i \leq 8$) and KI_i ($1 \leq i \leq 8$) to output a 64-bit ciphertext.

Describing FIG. 4 in more detail, an FL1 unit 410 encrypts the input 32-bit signal L_0 with an encryption key KL_1 and outputs a ciphertext L_1. An SLIMFO1 unit 510 encrypts the 32-bit ciphertext L_1 with encryption keys KO_1 and KI_1, outputs a signal SR_1 by encrypting the signal L_1 with the 32-bit signal R_0, and then outputs a signal R_1 by encrypting the signal SR_1 with encryption keys KO_2 and KI_2. An FL2 unit 420 encrypts the signal R_1 with an encryption key KL_2 and outputs a ciphertext R_2. An Exclusive-OR operation is performed to logically "exclusive OR" the signals L_0 and R_2 to provide a signal L_2 ($=SL_1$).

An FL3 unit 430 encrypts the signal L_2 with an encryption key KL_3 and outputs a ciphertext L_3. An SLIMFO2 unit 520 encrypts the signal L_3 with encryption keys KO_3 and KI_3, outputs a signal SR_2 by operating the encrypted signal L_3 with the signal SR_1, and then outputs a signal R_3 by encrypting the signal SR_2 with encryption keys KO_4 and KI_4. An FL4 unit 440 encrypts the signal R_3 with an encryption key KL_4 and outputs a ciphertext R_4. An Exclusive-OR operation is performed to logically "exclusive OR" the signals L_2 ($=SL_1$) and R_4 to provide a signal L_4 ($=SL_2$).

An FL5 unit 450 encrypts the signal L_4 with an encryption key KL_5 and outputs a ciphertext L_5. An SLIMFO3 unit 530 encrypts the signal L_5 with encryption keys KO_5 and KI_5, outputs a signal SR_3 by operating the encrypted signal L_3 with the signal SR_2, and then outputs a signal R_5 by encrypting the signal SR_3 with encryption keys KO_6 and KI_6. An FL6 unit 460 encrypts the signal R_5 with an encryption key KL_6 and outputs a ciphertext R_6. An Exclusive-OR operation is performed to logically "exclusive OR" the signals L_4 ($=SL_2$) and R_6 to provide a signal L_6 ($=SL_3$).

An FL7 unit 470 encrypts the signal L_6 with an encryption key KL_7 and outputs a ciphertext L_7. An SLIMFO4 unit 540 encrypts the signal L_7 with encryption keys KO_7 and KI_7, outputs a signal SR_4 by operating the encrypted signal L_7 with the signal SR_3, and then outputs a signal R_7 by encrypting the signal SR_4 with encryption keys KO_8 and KI_8. An FL8 unit 480 encrypts the signal R_7 with an encryption key KL_8 and outputs a ciphertext R_8. The signals L_6 (=SL_3) and R_8 are exclusive-OR-operated, resulting in a signal L_8 (=SL_4). Consequently, the eight FLi units ($1 \leq i \leq 8$) 410 to 480 and the four SLIMFOi units ($1 \leq i \leq 4$) 510 to 540 encrypt the 64-bit plaintext and output the 64-bit ciphertext, that is, the 32-bit signal SL_4//the 32-bit SR_4.

FIG. 5 is a block diagram illustrating an example of the SLIMFOi units illustrated in FIG. 4 according to an embodiment of the invention. Referring to FIG. 5, a SLIMFOi unit is an ith SLIMFO unit and implemented using parallel operations of signals in two FOi units. The SLIMFO1 unit 510 of FIG. 4 will be described by way of example. The SLIMFO1 unit 510 comprises an FO1 cipher 501 and an FO2 cipher 502. Each FO cipher includes $Fl_{i,j}$ sub-ciphers ($1 \leq i \leq 2$, $1 \leq j \leq 3$), for 3-round encryption.

The signal resulting from encrypting the 32-bit signal L_0 with the encryption key KL_1 in FIG. 4 is divided into a 16-bit signal L_0 (=L_1) and a 16-bit signal R_0 (=R_1) in the FO1 cipher 501. A signal L_2 is generated by performing a logical exclusive-OR operation on the signal L_1 with a sub-encryption key $KO_{1,1}$. An $Fl_{1,1}$ sub-cipher 511 generates a signal L_{2D} by encrypting the signal L_2 with a sub-encryption key $KI_{1,1}$. A delay (D6) 600 delays the signal R_1 and outputs a delayed signal R_{1D}. A signal L_3 is generated by performing a logical exclusive-OR operation on the signals R_{1D} and L_{2D}. Meanwhile, a signal R_2 is generated by performing a logical exclusive-OR operation on the signal R_1 with a sub-encryption key $KO_{1,2}$. An $FL_{1,2}$ sub-cipher 512 generates a signal R_{2D} by encrypting the signal R_2 with a sub-encryption key $KI_{1,2}$. A signal R_3 is generated

by performing a logical exclusive-OR operation on the signals R_{2D} and L_3. A signal L_4 is generated by performing a logical exclusive-OR operation on the signal L_3 with a sub-encryption key $KO_{1,3}$. An $Fl_{1,3}$ sub-cipher 513 generates a signal L_{4D} by encrypting the signal L_4 with a sub-encryption key $KI_{1,3}$. A delay (D7) 620 delays the signal R_3 and outputs a delayed signal R_{3D}. A 16-bit signal L_5 is generated by performing a logical exclusive-OR operation on the signals R_{3D} and L_{4D}.

The 32-bit signal R_0 which was divided from the 64-bit signal in FIG. 4 is further divided into a 16-bit signal $L_{0'}$ and a 16-bit signal $R_{0'}$ in the FO2 cipher 502. A signal L_6 is generated by performing a logical exclusive-OR operation on the signal $L_{0'}$ using the 16-bit signal L_5. Meanwhile, a signal R_4 is generated by performing a logical exclusive-OR operation on the signal $R_{0'}$ using the 16-bit signal R_3. A signal R_5 is generated by performing a logical exclusive-OR operation on the signal R_4 using a sub-encryption key $KO_{2,1}$. An $Fl_{2,1}$ sub-cipher 514 generates a signal R_{5D} by encrypting the signal R_5 with a sub-encryption key $KI_{2,1}$. A signal R_6 is generated by performing a logical exclusive-OR operation on the signals R_{5D} and L_6. That is, the $Fl_{1,3}$ sub-cipher 513 and the $Fl_{2,1}$ sub-cipher 514 synchronize the signal L_6 to the signal R_6 without using delays. A signal L_7 is generated by performing a logical exclusive-OR operation on the signal L_6 with a 16-bit sub-encryption key $KO_{2,2}$. An $FL_{2,2}$ sub-cipher 515 generates a signal L_{7D} by encrypting the signal L_7 with a 16-bit sub-encryption key $KI_{2,2}$. A delay (D8) 640 delays the signal R_6 and outputs a delayed signal R_{6D}. A signal L_8 is generated by performing a logical exclusive-OR operation on the signals L_{7D} and R_{6D}. A signal R_7 is generated by performing a logical exclusive-OR operation on the signal R_6 with a 16-bit sub-encryption key $KO_{2,3}$. An $Fl_{2,3}$ sub-cipher 516 generates a signal R_{7D} by encrypting the signal R_7 with a 16-bit sub-encryption key $KI_{2,3}$. A signal R_8 is generated by performing a logical exclusive-OR operation on the signals R_{7D} and L_8. Consequently, a 32-bit ciphertext $L_8 \| R_8$ is generated by operating the 16-bit signal L_8 with the 16-bit signal R_8.

As described above, the SLIMFO1 unit encrypts the input plaintext by processing the 16-bit signals L_0 and R_0 in parallel in the FO1 cipher 501 and processing the 16-bit signals $L_{0'}$ and $R_{0'}$ in parallel in the FO2 cipher 502. The parallel processing of the 32-bit signals L_0 and R_0 which were divided from the 64-bit input signal in the SLIMFOi units remarkably increases encryption speed and reduces the number of delays used to synchronize a delayed signal to a non-delayed signal.

FIG. 6 is a block diagram illustrating an example of the $Fl_{i,j}$ sub-ciphers illustrated in FIG. 5 according to an embodiment of the invention. By way of example, the $Fl_{1,1}$ sub-cipher 511 will be described below.

Referring to FIG. 6, the $Fl_{1,1}$ sub-cipher 511 includes a first ciphering unit and a second ciphering unit. In the first ciphering unit, a 16-bit input signal is divided into a 9-bit signal RL_0 and a 7-bit signal RR_0. An S91 operator 710 generates a 9-bit signal y0, y1, . . ., y8 from the input signal RL_0 by

```
y0 = (x0x2)⊕x3⊕(x2x5)⊕(x5x6)⊕(x0x7)⊕(x1x7)⊕(x2x7)⊕(x4x8)⊕(x5x8)⊕(x7x8)⊕'1';
y1 = x1⊕(x0x1)⊕(x2x3)⊕(x0x4)⊕(x1x4)⊕(x0x5)⊕(x3x5)⊕x6⊕(x1x7)⊕(x2x7)⊕(x5x8)⊕'1';
y2 = x1⊕(x0x3)⊕(x3x4)⊕(x0x5)⊕(x2x6)⊕(x3x6)⊕(x5x6)⊕(x4x7)⊕(x5x7)⊕(x6x7)⊕x8⊕(x0x8)⊕'1';
y3 = x0⊕(x1x2)⊕(x0x3)⊕(x2x4)⊕x5⊕(x0x6)⊕(x1x6)⊕(x4x7)⊕(x0x8)⊕(x1x8)⊕(x7x8);
y4 = (x0x1)⊕(x1x3)⊕x4⊕(x0x5)⊕(x3x6)⊕(x0x7)⊕(x6x7)⊕(x1x8)⊕(x2x8)⊕(x3x8);
y5 = x2⊕(x1x4)⊕(x4x5)⊕(x0x6)⊕(x1x6)⊕(x3x7)⊕(x4x7)⊕(x6x7)⊕(x5x8)⊕(x6x8)⊕(x7x8)⊕'1';
y6 = x0⊕(x2x3)⊕(x1x5)⊕(x2x5)⊕(x4x5)⊕(x3x6)⊕(x4x6)⊕(x5x6)⊕x7⊕(x1x8)⊕(x3x8)⊕(x5x8)⊕(x7x8);
y7 = (x0x1)⊕(x0x2)⊕(x1x2)⊕x3⊕(x0x3)⊕(x2x3)⊕(x4x5)⊕(x2x6)⊕(x3x6)⊕(x2x7)⊕(x5x7)⊕x8⊕'1';
y8 = (x0x1)⊕x2⊕(x1x2)⊕(x3x4)⊕(x1x5)⊕(x2x5)⊕(x1x6)⊕(x4x6)⊕x7⊕(x2x8)⊕(x3x8);
```

. . . . (3)

That is, the S91 operator 710 generates the 9-bit signal y1, y2, . . ., y8 by

performing parallel logical AND operations and then performing a logical exclusive-OR operation of a 9-bit signal x0, x1, . . ., x8 in parallel. A ZE1 unit 720 receives the signal RR_0, adds two zeroes to the MSB of the signal RR_0, and outputs a 9-bit signal. An Exclusive-OR operation is performed to logically "exclusive OR" the outputs of the S91 operator 710 and the ZE1 unit 720 to provide a 9-bit signal RL_1. Another Exclusive-OR operation is performed to logically "exclusive OR" the signal RL_1 and a 9-bit sub-encryption key $KI_{1,1,2}$, to provide a 9-bit signal RL_2. The signal RL_2 is temporarily stored in a first register (register 1) 800.

Simultaneously, an S71 operator 740 generates a 7-bit signal y0, y1, . . ., y6 from the input signal RR_0 (=RR_1) by

y0=(x1x3)⊕x4⊕(x0x1x4)⊕x5⊕(x2x5)⊕(x3x4x5)⊕x6⊕(x4x6)⊕(x1x6)⊕(x3x6)⊕(x2x4x6)⊕(x1x5x6)⊕(x4x5x6);
y1=(x0x1)⊕(x0x4)⊕(x2x4)⊕x5⊕(x1x2x5)⊕(x0x3x5)⊕x6⊕(x0x2x6)⊕(x3x6)⊕(x4x5x6)⊕1;
y2=x0⊕(x0x3)⊕(x2x3)⊕(x1x2x4)⊕(x0x3x4)⊕(x1x5)⊕(x4x5)⊕(x0x5)⊕(x0x4x6)⊕(x2x6)⊕(x4x6)⊕1;
y3=x1⊕(x0x1x2)⊕(x1x4)⊕(x3x4)⊕(x0x5)⊕(x0x1x5)⊕(x2x3x5)⊕(x1x4x5)⊕(x2x6)⊕(x1x3x6);
y4=(x0x2)⊕x3⊕(x1x3)⊕(x1x4)⊕(x0x1x4)⊕(x2x3x4)⊕(x1x5)⊕(x1x3x5)⊕(x0x4x5)⊕(x1x6)⊕(x3x6)⊕(x0x3x6)⊕(x5x6)⊕1;
y5=x2⊕(x0x2)⊕(x0x3)⊕(x1x2x3)⊕(x0x2x4)⊕(x1x5)⊕(x2x5)⊕(x1x5)⊕(x1x6)⊕(x1x2x6)⊕(x0x3x6)⊕(x3x4x6)⊕(x2x5x6)⊕1;
y6=(x1x2)⊕(x0x1x3)⊕(x0x4)⊕(x1x5)⊕(x3x5)⊕x6⊕(x3x4x6)⊕(x2x3x6)⊕(x1x4x6)⊕(x4x5x6);

. . . . . (4)

That is, the S71 operator 740 generates the 9-bit signal y1, y2, . . ., y6 by performing parallel logical AND operations and then performing a logical exclusive-OR operation of a 7-bit signal x0, x1, . . ., x6 in parallel. A TR1 unit 730 removes two zeroes from the MSBs of the 9-bit signal RL_1 and outputs the resulting 7-bit signal. A 7-bit signal RR_2 is generated by performing a logical exclusive-OR operation on the outputs of the TR1 730 and the S71 operator 740 with a sub-encryption key $KI_{1,1,1}$. The signal RR_2 is temporarily stored in the first register 800. Upon receipt of a first clock signal CLK1 from a controller (not

shown), the register 800 simultaneously outputs the 9-bit signal RL_2 and the 7-bit signal RR_2. Thus the register 800 functions to synchronize the output timings of signals according to delay involved with encryption in the S91 operator 710, the ZE1 unit 720, the TR1 unit 730, and the S71 operator 740.

In the second ciphering unit, an S92 operator 750 generates a 9-bit signal y0, y1, . . ., y8 from the 9-bit signal RL_2 received from the register 800 using Eq. (3). A ZE2 unit 760 adds two zeroes to the MSB of the signal RR_2 received from the register 800 and outputs a 9-bit signal. An Exclusive-OR operation is performed to logically "exclusive OR" the outputs of the S92 operator 750 and the ZE2 unit 760 to provide a 9-bit signal RL_3. The signal RL_3 is temporarily stored in a second register (register 2) 820.

Simultaneously, an S72 operator 780 generates a 7-bit signal y0, y1, . . ., y6 from the 7-bit signal RR_2 (=RR_3) using Eq. (4). A TR2 unit 770 removes two zeroes from the MSBs of the 9-bit signal RL_3 and outputs the resulting 7-bit signal. A 7-bit signal RR_4 is generated by performing a logical exclusive-OR-operation on the outputs of the TR2 770 and the S72 operator 780. The signal RR_4 is temporarily stored in the second register 820.

Upon receipt of a second clock signal CLK2 from the controller, the register 820 simultaneously outputs the 9-bit signal RL_4 and the 7-bit signal RR_4. Thus the register 820 functions to synchronize the output timings of signals according to the delay involved with the encryption in the S92 operator 750, the ZE2 unit 760, the TR2 unit 770, and the S72 operator 780.

As described above, the S91 operator 710 and the S92 operator 750 each output a 9-bit signal y0, y1, . . ., y8 by performing parallel logical AND operations and then performing a logical exclusive-OR operation according to Eq. (3). The S71 operator 740 and the S72 operator 780 each output a 7-bit signal

y0, y1, . . ., y6 by parallel AND operations and then exclusive-OR operation according to Eq. (4). Therefore, encryption speed is remarkably increased. Furthermore, the use of the registers 800 and 820 for signal timing synchronization enables output of an accurate ciphertext.

In accordance with the present invention, (1) parallel computation of input signals increases signal processing speed; (2) due to synchronization of the output timings of a delayed signal and a non-delayed signal, an accurate ciphertext is achieved and thus an encryption system is further stabilized; and (3) the decrease in devices used for synchronization reduces required chip capacity and production cost.

While the invention has been shown and described with reference to a certain embodiment thereof, it will be understood by those skilled in the art that various changes in form and details may be made therein without departing from the spirit and scope of the invention as defined by the appended claims.